

# The Accurate Particle Tracer Code

Yulei Wang,<sup>1,2</sup> Jian Liu,<sup>1,2,\*</sup> Hong Qin,<sup>1,3</sup> and Zhi Yu<sup>4,5</sup>

<sup>1</sup>*School of Nuclear Science and Technology and Department of Modern Physics,  
University of Science and Technology of China, Hefei, Anhui 230026, China*

<sup>2</sup>*Key Laboratory of Geospace Environment, CAS, Hefei, Anhui 230026, China*

<sup>3</sup>*Plasma Physics Laboratory, Princeton University, Princeton, NJ 08543, USA*

<sup>4</sup>*Theory and Simulation Division, Institute of Plasma Physics,  
Chinese Academy of Sciences, Hefei, Anhui 230031, China*

<sup>5</sup>*Center for Magnetic Fusion Theory,  
Chinese Academy of Sciences, Hefei, Anhui 230031, China*

## Abstract

The Accurate Particle Tracer (APT) code is designed for large-scale particle simulations on dynamical systems. Based on a large variety of advanced geometric algorithms, APT possesses long-term numerical accuracy and stability, which are critical for solving multi-scale and non-linear problems. Under the well-designed integrated and modularized framework, APT serves as a universal platform for researchers from different fields, such as plasma physics, accelerator physics, space science, fusion energy research, computational mathematics, software engineering, and high-performance computation. The APT code consists of seven main modules, including the I/O module, the initialization module, the particle pusher module, the parallelization module, the field configuration module, the external force-field module, and the extendible module. The I/O module, supported by Lua and Hdf5 projects, provides a user-friendly interface for both numerical simulation and data analysis. A series of new geometric numerical methods and key physical problems, such as runaway electrons in tokamaks and energetic particles in Van Allen belt, have been studied using APT. As an important realization, the APT-SW version has been successfully distributed on the world's fastest computer, the Sunway TaihuLight supercomputer, by supporting master-slave architecture of Sunway many-core processors.

---

\* corresponding author: jliuphy@ustc.edu.cn

## I. INTRODUCTION

Nonlinear and multi-scale dynamical processes are ubiquitous in different fields of scientific and engineering researches. Especially in plasma physics, where long-range collective phenomena dominate, advanced numerical schemes and powerful computing software are required for solving complex physical and technical problems. The GeoAlgorithmic Plasma Simulator (GAPS) project is initiated in order to solve various difficult yet key problems in plasma-related domains by applying advanced geometric algorithms and modern large-scale simulation techniques. The Accurate Particle Tracer (APT) code is one product of the GAPS project aiming to large-scale particle simulations on dynamical systems.

Although fruitful results have been achieved by computer simulation studies, numerical errors still hamper the application and validation of simulations. The accumulation of numerical errors makes traditional algorithms unreliable when dealing with multi-scale and non-linear processes and carrying out long-term simulations. Real physical information may be distorted due to the breakdown of original physical structures in computation schemes. To guarantee long-term numerical accuracy and stability, a series of advanced geometric algorithms have been systematically developed recently [1–10], which bound the global numerical errors by preserving geometric structures and promote the simulation ability markedly. On the other hand, the application of latest geometric algorithms and the implementation of efficient and powerful simulation tools require interdisciplinary study and cooperation. A universal platform for experts in different fields needs to be designed to integrate the latest trans-disciplinary achievements efficiently. Under the well-designed integrated and modularized framework, the APT code serves as a universal platform for researchers from different fields, such as plasma physics, accelerator physics, space science, fusion energy research, computational mathematics, software engineering, and high-performance computation. Based on a large variety of advanced geometric algorithms [1–14], APT possesses long-term numerical accuracy and stability, which are critical for solving multi-scale and non-linear problems.

The underlying model of APT is the first principle particle model. In simulations of plasma systems, the distribution of plasma particles are sampled statistically in the phase space. APT traces each sampling particle accurately by solving classical or relativistic Lorentz force equations that are the characteristic line equations of the Vlasov equation,

using appropriate geometric algorithms. If considering the collisional effects, bremsstrahlung radiation, or source terms, numerical stochastic differential equations are employed during particle tracing. The external electromagnetic fields are set up through analytical functions, which usually come from known theoretical models, or direct field configuration data, which may be obtained from experimental measurements or other simulation results. There already exist many built-in electromagnetic field configurations for typical physical problems in APT, which can be conveniently used for physical research and numerical analysis. Self-consistent fields are also available in some models by calculating mean fields from particle distribution in the phase space.

The APT code is implemented in standard C-language and can be distributed on Linux as well as many other operation systems conveniently. The APT code consists of seven main modules, including the I/O module, the initialization module, the particle pusher module, the parallelization module, the field configuration module, the external force-field module, and the extendible module. The I/O module calls the libraries from Lua and Hdf5 projects. The input configuration files consist of several Lua scripts which make it convenient for users to set the parameters of physical problems. The Hdf5 standard format is applied to output data, which enables users to access data in a file-system-like way, unifies the data format, and simplifies data analysis. The initialization module defines the data structure of particle and provides a number of methods for statistical sampling. The particle pusher module contains a great deal of pushers based on advanced geometric algorithms, including volume-preserving and symplectic integrators with different orders and stability regions. Many traditional algorithms such as Runge-Kutta method with different orders are also available. It is thus convenient to choose appropriate algorithms for solving realistic problems and studying numerical methods under complex physical setup. The parallelization module realizes the parallelization of the APT code in several ways and different computation environments. In large-scale parallel computations, the calculation of dynamics of massive sampling particles is assigned to different processors or cores by the parallelization module. The field configuration module and the external force-field module contain respectively various electromagnetic configurations and other external force fields, such as radiation force, collisional force, bremsstrahlung force, and gravitation field. Written by the bash-script of Linux system, the extendible module provides a convenient way to extend the source code. After adding new global variables, algorithms, field configurations,

and external forces into corresponding files, one can easily modify the APT source code and obtain a new version suitable for a specific task, which can be compiled directly through rewriting and running a single bash script.

With the aid of APT, a series of new geometric numerical methods have been studied systematically. Many brand new and influential physical results have been discovered accordingly [11, 13]. In this paper, we present two typical cases to exhibit the advantages and achievements of APT. In the first case, the multi-scale dynamics of runaway electrons in tokamaks is studied by secular simulations over  $10^{12}$  time steps. Using the relativistic volume-preserving algorithm in APT, the fine structures of runaway transit orbits are revealed. The APT code shows tremendous long-term numerical accuracy and enables the discovery of novel mechanisms such as neoclassical collisionless pitch-angle scattering and new runaway energy limit rule. In the second case, the evolution of energetic particle distribution in Van Allen belt is simulated by tracing massive sampling particles by APT. The Euler-Symplectic pusher for relativistic dynamics is employed. The diffusion process of energetic particles in the terrestrial magnetic field is recovered precisely.

As an important realization, a branch version APT-SW for the world's fastest supercomputer, the Sunway TaihuLight supercomputer [15], has been released recently. The Sunway TaihuLight provides more than ten million computation cores and has the peak performance of 125PFlops [16]. To fully utilize the computation ability of the Sunway TaihuLight, APT-SW is designed to support the Sunway many-core architecture. Distributed successfully on the Sunway TaihuLight platform, APT-SW is used to carry out the largest particle simulation with more than  $10^{21}$  floating-point operations. Supported by the powerful computation capacity, APT-SW can be used to implement more realistic simulations with large amount of sampling particles, more complex field configurations, and multi-timescales spanning 10 orders. Meanwhile, other versions of APT for GPU acceleration are also available.

The rest part of the paper is organized as follows. The structure and modules of APT is introduced in Sec. II. In Sec. III, two typical simulation cases carried out by APT are exhibited. The realization of the branch version APT-SW and its distribution on the Sunway TaihuLight supercomputer is discussed in Sec. IV. Section V provides a brief summary.

## II. ARCHITECTURE OF APT

As a product of the GAPS project, APT emphasizes the development and application of advanced geometric algorithms in various simulation researches to enhance the accuracy and reliability. To facilitate the rapid and professional application of simulation methods and techniques, the architecture of APT is designed to be integrated, modularized, and extendible. Figure 1 depicts the design schematic of the APT code. There are four containers for configuration parameters, electromagnetic functions, external forces, and particle pushers respectively. Through the extendible module, one can conveniently add new variables, electromagnetic field configurations, external forces, as well as new algorithms to these containers, see the brown arrows in Fig. 1. Several newly developed advanced geometric algorithms have been built into the algorithm container. The flow process of APT is denoted by the blue arrows in Fig. 1. Users can set all required simulation parameters in a single Lua configuration file, which picks up appropriate parameters and loads them to the global variable space of APT. Based on the global variables, the procedure selects corresponding electromagnetic functions, external forces, and algorithms from containers. Some global variables are also called during simulation initialization, pusher iteration, and data output. After configuring the output environment, APT initializes all the sampling particles and assigns the tasks to corresponding processors by the parallelization module. The phase space states of sampling particles are pushed iteratively by the selected algorithm, and the required data flow is outputted to the file system. More details of each APT module are introduced as follows.

The particle pusher module governs the kernel of APT computation. It contains various geometric algorithms including volume-preserving and symplectic algorithms with different orders and stabilities for both classical and relativistic systems. The volume-preserving algorithms (VPA) include both non-relativistic and relativistic VPAs of different orders [3, 5, 9, 10]. VPAs ensure the conservation of phase space volume, the intrinsic geometric attribute of Lorentz force system [10]. The symplectic algorithms built in APT contain the canonical Euler-symplectic algorithm, the implicit mid-point symplectic algorithm, and the K-symplectic algorithm [17, 18]. Canonical or non-canonical symplectic structures are preserved during iteration by these symplectic algorithms. Through the preservation of geometric structures, the geometric algorithms in the particle pusher module equip APT

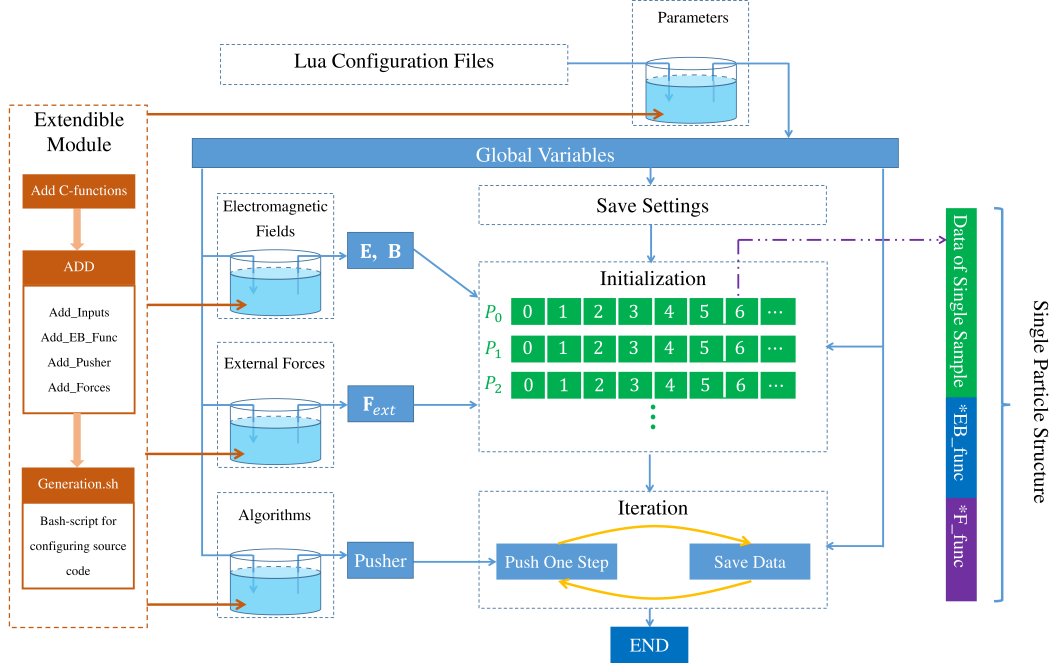


Figure 1. Diagram of the APT architecture. The blue arrows denote the flow of procedure, while the brown parts represent the extendible module. In the Initialization box, sampling particles assigned to different processors are marked by  $P_0, P_1, P_2, \dots$ .

with long-term stability. For example, it has been verified that VPAs can trace the dynamics of particle correctly for tens of billion steps [5, 9–11, 13]. Besides the geometric algorithms, traditional algorithms, such as the Runge-Kutta algorithms of different orders, are also available in APT. These traditional algorithms can be used as references for the analysis of new algorithms.

Libraries of Lua and Hdf5 projects support the I/O module of APT. The Lua libraries provide the APIs of C-language which help one to load variables of different types, including floating-point numbers, integer numbers, strings, and functions, into the main procedure from Lua scripts [19]. In APT, parameters needed by each problem are organized in one Lua script. For example, the Lua script for charged particles in tokamaks sets all the necessary parameters, such as the major radius and the minor radius of the tokamak device, the configuration of electromagnetic fields, the initial conditions of particles. Through modifying the Lua script, researchers can easily manage the studies of one corresponding problem, which improves the efficiency of researches. APT also provides an introduction file for all the setting parameters, through which people can write their own Lua scripts and extend

the applications of APT. All the necessary data and the setting parameters are recorded in a single Hdf5 format file. The Hdf5 libraries provide a file-system-like way of reading and writing data [20], i.e., people can access the data like manipulating a file in Linux operation system, which simplifies the data analysis. Supported by most of the popular data-analysis tools, such as Matlab and Python, the standard Hdf5 format helps unify the format of data. Based on Lua and Hdf5 projects, the I/O module provides a user-friendly and efficient interface for both numerical simulation and data analysis.

The initialization module defines the data structure of particles and initializes particle statistical samples. The particle structure, see Fig. 1, contains basic information of a particle sample, including time, position, momentum, acceleration, electromagnetic field, external forces, and works of forces. Selected function pointers of both electromagnetic field and other external forces are also parts of the particle structure, which are determined by the global variables loaded from the Lua configuration files. Correspondingly, through the particle structure, people can access all the data and functions needed for implementing computations. The initialization module also provides several common-used methods of statistical sampling, such as uniform sampling and normal-distribution sampling, which facilitate APT for dealing with complex initial conditions.

As a vital part of large-scale simulations, the parallelization module assigns all the particle samples to different processors or cores and controls the simulations in two optional modes, namely, the asynchronous mode and the synchronous mode. The group of samples is divided into many parts according to the available calculation processors,  $P_0, P_1, \dots$ . The asynchronous parallelization simulates particles independently and can be applied to problems with ignorable self-consistent fields. If a processor is assigned more than one particles, see  $P_0$  in Fig. 1, APT would not push the second particle until the simulation of the first one is finished. This parallelization mode minimizes the communications between different processors, improves the efficiency, and makes it easy to distribute APT on supercomputers with different environments. For cases with the self-consistent fields, the synchronous parallelization is required. Under this mode, particle samples are pushed synchronously and the information of self-consistent fields of all samples can be collected at given moments.

In the field configuration module, there are various build-in electromagnetic configurations, including the tokamak field, the terrestrial magnetic field, the uniform field, the radially non-uniform magnetic field, and the electric oscillator field. Besides the electromagnetic

function, the scalar and vector potentials are also available in order to feed some geometric algorithms. The field configuration module also provides an interface for inputting discrete field data obtained from experiments or simulations. Users should organize the discrete data in a file with the standard format and write a boundary-test function which indicates the region of data. Then the discrete data can be loaded directly by APT, and be interpolated to obtain the continuous field used to push particles.

The external force-field module adds non-electromagnetic forces into the dynamical simulation of particles. Various external forces including synchrotron radiation force, collisional force, bremsstrahlung force, and gravitation field have been built in APT code. Different external forces are calculated in different ways to ensure the accuracy. The synchrotron radiation and the gravitation field are treated as effective field in particle pushers. Even though the geometric structure of Lorentz system is modified, the geometric algorithms can still guarantee secular stability due to the preservation of electromagnetic geometric structure [11, 13]. To calculate random forces, such as collisional force and bremsstrahlung force, APT also provides numerical stochastic differential equations pushers to give exquisite random simulation. By use of the external force-field module of APT, one can study complex physical systems in many different fields.

The extendible module of APT is implemented by Bash-script of Linux operation system. Through three steps, users can conveniently build a new version of APT for new applications, see Fig. 1. First, users should add new global variables or new C-functions of algorithms, electromagnetic fields, and external forces in corresponding modules of APT. Then, they should modify a file named “ADD” and tell the program the names, types and introductions of these new variables or functions by use of several bash-functions including “Add\_Inputs”, “Add\_EB\_Func”, “Add\_Pusher” and “Add\_Forces”. Finally, through running the script named “Generation.sh”, all the source codes and introduction files corresponding to the extended parts will be generated automatically. The extended version of APT can be compiled directly. Users should also write a Lua configuration file to test all the changes and to do their own projects. By use of the extendible module, researchers from different fields can conveniently focus on their own tasks and use the components provided by others directly. For example, researchers from the plasma physics, accelerator physics, space science, and fusion energy can add the customized electromagnetic configurations and directly use the advanced geometric algorithms provided by computational mathematicians.



The complex physical field configurations can be used by mathematicians to analyze their newly developed algorithms. All of these physical and mathematical cases can provide various test environments for efficiency optimization to the experts from software engineering and high-performance computation, which is helpful to improve the performances of APT. This extendible module can thus boost the integration of new results from different fields.

### III. APPLICATION CASES OF APT

The APT code has been used in the studies of new algorithms and the simulations of a variety of important plasma processes. Due to the secular stability provided by the geometric algorithm kernel, APT can solve many multi-scale and nonlinear problems that cannot be addressed by traditional simulation methods, which stimulates the discoveries of new physical phenomena. In this section, we exhibit the prospects of APT in two applications, namely, the secular dynamics of runaway electron in tokamak and the distribution evolution of energetic particles in Van Allen belts.

#### A. The Secular Dynamics of Runaway Electrons

Runaway electrons are energetic charged particles generated in tokamaks which are the most prospective devices for controlled fusion energy [21, 22]. During the operation of tokamaks, fast shutdown, disruptions, and strong current drive can induce the generation of large amounts of runaway electrons [23–37]. Due to the acceleration of the induced loop electric field, runaway electrons carrying energies of tens of MeVs have been reported in various experiments [38–41]. The existence of runaway electrons is a potential threat to safety operation of tokamak devices. To study the runaway process, the relativistic effect and the synchrotron radiation cannot be neglected. The runaways reach the synchrotron energy limit when the effect of the electric field is balanced out by the synchrotron radiation dissipation[4, 12, 42, 43]. Both the detailed dynamical behaviors and the rules of energy limit of runaway electrons are important topics in the field of fusion energy.

The runaway dynamics is a typical multi-scale process, which involves timescales from characteristic time of Lorentz force ( $10^{-11}$  s) to energy balance time (1 s) [12, 13]. The runaway dynamics thus spans about 11 orders of magnitude in timescale. To eliminate the

trouble of multi-timescale, traditional methods average out the motion in small timescale, which simplifies the problem but losses physical information. However, with the secular stability of geometric algorithms, APT can simulate directly the full-orbit dynamics which needs more than  $10^{12}$  iteration steps. Correspondingly, new and fine pictures of runaway dynamics in different timescales have been found. Especially, the discovery of neoclassical collisionless pitch-angle scattering brings a novel understanding about runaway behaviors and results in new laws of energy limit [11, 13].

Figure 2 depicts the snapshots of runaway orbit projected on poloidal plane at different moments simulated by APT. This full-orbit result is different from results of traditional studies and exhibits clearly the fine ripple structures in small timescale [4, 13]. To obtain Fig.2, we use the relativistic volume-preserving algorithm in APT. Because the runaway dynamics involves the synchrotron radiation, the external force-field module of APT is also used. The tokamak field configuration file of APT provides the needed field given by

$$\mathbf{B} = -\frac{B_0 R_0}{R} \mathbf{e}_\xi - \frac{B_0 \sqrt{(R - R_0)^2 + z^2}}{qR} \mathbf{e}_\theta, \quad (1)$$

$$\mathbf{E} = E_l \frac{R_0}{R} \mathbf{e}_\xi, \quad (2)$$

where,  $\mathbf{e}_\xi$  and  $\mathbf{e}_\theta$  are respectively the toroidal and poloidal unit vectors,  $R_0$  is the major radius,  $q$  denotes safety factor,  $E_l$  is the strength of loop electric field, and  $B_0$  is the magnitude of background magnetic field. In calculation, we set parameters based on a typical tokamak, that is  $R_0 = 1.7$  m,  $a = 0.4$  m,  $q = 2$ ,  $B_0 = 2$  T, and  $E_l = 0.2$  V/m. The initial position of a runaway eletron is chosen as  $R = 1.8$  m,  $\xi = z = 0$ , and the initial parallel and perpendicular momentums are set as  $p_{\parallel 0} = 5 m_0 c$  and  $p_{\perp 0} = 1 m_0 c$  respectively. The time step of simulation is set as  $\Delta t = 1.9 \times 10^{-12}$  s. The total number of iteration steps is thus about  $1.6 \times 10^{12}$ . The long-term stability of APT ensures the correctness of simulation results even after  $10^{12}$  steps.

## B. Energetic Particles in Van Allen Radiation Belt

Since its discovery in 1958, the Van Allen radiation belts have been an important topic of geophysics [44]. The Van Allen belt is a region about several  $R_E$ s away from the surface of earth, where  $R_E$  is the radius of earth at the equator. Large amounts of particles carrying

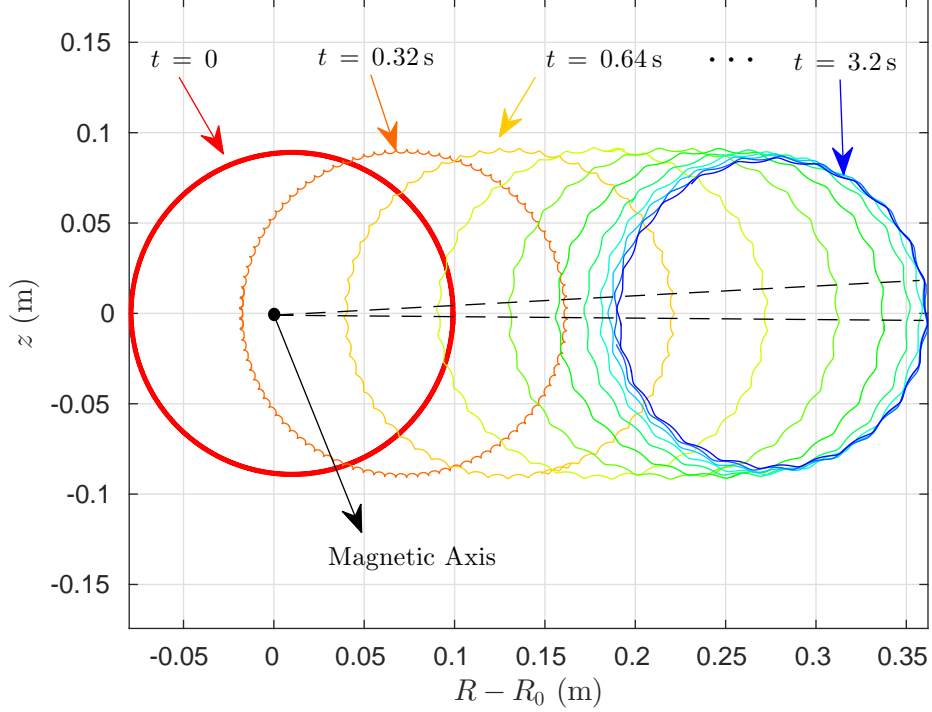


Figure 2. Full-orbit snapshots of runaway orbit projected in the poloidal plane at different moments. The configuration of field is determined by  $R_0 = 1.7\text{ m}$ ,  $a = 0.4\text{ m}$ ,  $q = 2$ ,  $B_0 = 2\text{ T}$ , and  $E_l = 0.2\text{ V/m}$ . The runaway electron is initially sampled with momentum  $p_{\parallel 0} = 5\text{ m}_0c$ ,  $p_{\perp 0} = 1\text{ m}_0c$  at  $R = 1.8\text{ m}$ ,  $\xi = z = 0$ . Besides the neoclassical radial drift, the ripple structures are obviously exhibited.

energies of several MeVs are confined in the radiation belt. These energetic particles do harm to the spacecraft as well as the satellites. The mechanism of acceleration of energetic particles in radiation belts is still not clear, even though many works have been done on this problem [45, 46].

APT has offered an outstanding platform for studying the statistical behaviors and the acceleration mechanisms of radiation-belt particles. The background terrestrial field is approximated by

$$\mathbf{B}(\mathbf{x}) = \frac{B_0 R_0^3 z}{(R^2 + z^2)^2} \mathbf{e}_R + \frac{B_0 R_0^3}{R^2 + z^2} \left( -\frac{R}{R^2 + z^2} + \frac{1}{2R} \right) \mathbf{e}_z.$$

The corresponding vector field is

$$\mathbf{A}(\mathbf{x}) = \frac{B_0 R_0^3}{2(R^2 + z^2)} \mathbf{e}_\theta, \quad (3)$$

where  $R_0 = 6.6R_E$  and  $B_0 = 2B_{surf}/6.6^3$ ,  $R_E = 6370000$  m is the radius of earth, and  $B_{surf} = 3.12 \times 10^{-5}$  T is the magnetic field on surface of earth. The parallelization module is used to achieve large-scale statistical simulations on supercomputers. People can choose different geometric algorithms, adopt various initial statistical samplings, and test different electromagnetic-wave acceleration processes in APT.

For example, we choose the Euler-Symplectic algorithm for relativistic charged particles to simulate the collective evolution of energetic electrons in the background magnetic field of earth. Initially, we uniformly sample  $10^4$  particles in the region  $3.3R_E \leq R \leq 3.7R_E$  and  $-0.2R_E \leq z \leq 0.2R_E$ . The initial Lorentz factors of particles are given by a normal distribution with the average value  $\mu = 0.6$  MeV and the standard deviation  $\sigma = 0.5$  KeV, and the initial momentums are sampled uniformly within  $-1 \leq p_{\parallel}/p \leq 1$ . The time step is about  $5.24 \times 10^{-6}$  s, and the number of steps is  $2.8 \times 10^5$ . The diffusion process of energetic electrons in Van Allen belt of earth is shown in Fig. 3. Since the confined particles in the initial region have different momentums and phases, they do not move together but spread in the earth magnetic field. The profile of distribution at  $t = 0.15$  s is similar to the shape of radiation belts, which shows the correctness of the simulation.

#### IV. APT ON SUNWAY TAIHULIGHT SUPERCOMPUTER

Recently, the Top1 supercomputer in the world has been updated after the construction of the Sunway TaihuLight supercomputer in Wuxi, China. Possessing peak performance of 125PFlops, the Sunway TaihuLight becomes the fastest computer in the world [15, 16]. The calculation capability of Sunway TaihuLight is provided by 40000 SW26010 many-core processors, each of which has 260 processing elements [16]. Therefore, there are more than  $10^7$  processors available on this supercomputer. The Sunway TaihuLight supercomputer has a total storage of 20PB, which makes it possible for the storing and analysis of big data. However, it is not straightforward to distribute a program on this supercomputer, because the architecture of the processor is different from any other CPUs or GPUs. In each SW26010 processor, four master cores and 256 slave cores are integrated. The two types of cores have similar Flops but different architectures. A program should utilize both the master and slave cores to avoid the waste of the calculation capability of the Sunway TaihuLight. Meanwhile, the communication between a master core and its slave cores becomes the bottleneck of

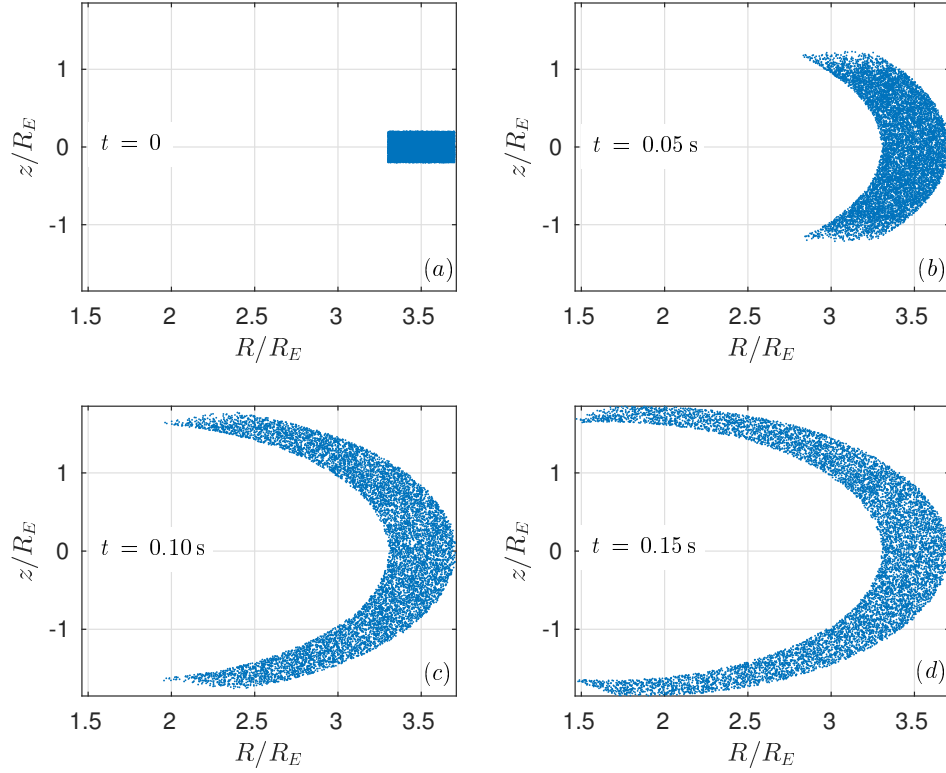


Figure 3. The diffusion process of energetic electrons in Van Allen belts.

efficiency because the speed of accessing shared memory for slave cores is quite slow. To improve the memory bandwidth, the SW26010 processor has a 64K local cache memory, also called local data memory (LDM), for slave cores. The slave cores can access the data stored in the LDM with high speed. Consequently, to improve the efficiency, one should appropriately use the LDM to reduce the access of data in shared memory by the slave cores and to minimize the communication between master and slave cores. To achieve these goals, the program should be supported by the master-slave acceleration libraries of the Sunway TaihuLight platform.

As a branch version of the APT code, APT-SW has been successfully distributed on the Sunway TaihuLight supercomputer. APT-SW remains most of the attributes of APT while several modifications have been done to adopt the computation environment of the Sunway TaihuLight. To implement the master-slave acceleration, the APT-SW source code requires both the master-core and the slave-core commands. The master-core part is in charge of the data I/O, the simulation initialization, the process parallelization, while the slave-core part

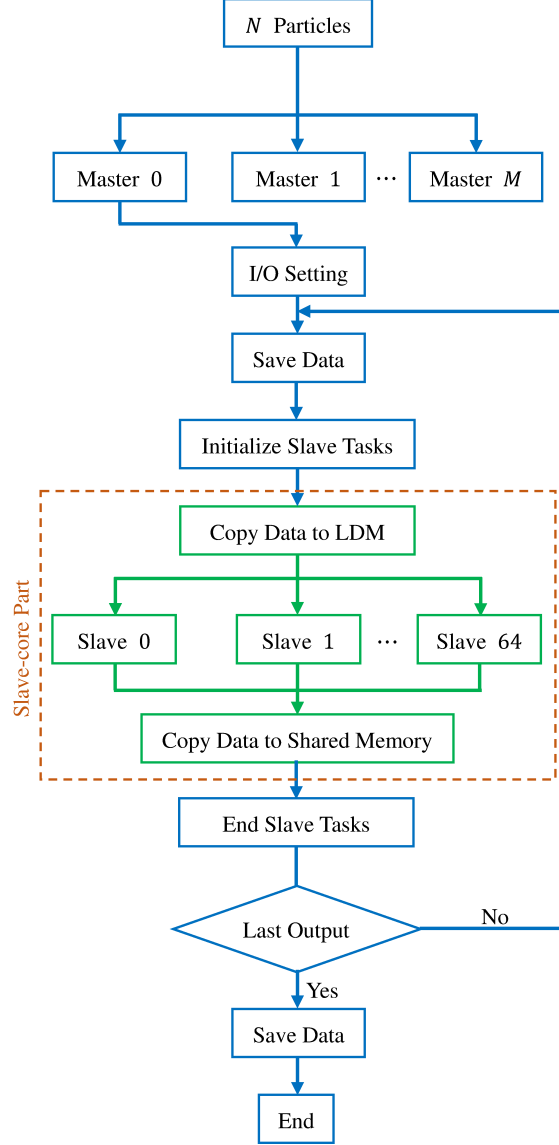


Figure 4. Schematic of master-slave acceleration of APT-SW on the Sunway TaihuLight supercomputer. The total number of particle samplings is  $N = 64 \times M$ , where  $M$  is the number of master processes. Because the data can only be written into the file system through master processors and the efficiency of communication between master and slave cores is low,  $n$  iterations are divided into  $m$  groups based on the users' output configuration, and the data saving executes for  $m$  times. In common plasma simulations,  $n$  is much larger than  $m$ . Therefore, the time of master-slave communications can be neglected compared with the iteration time. In the brown dashed box, the slave part of APT is depicted. Between two processes of data saving, the master process asks each slave core to execute iteration of  $n/m$  times.

contains the particle pusher with geometric algorithms, the electromagnetic field module, the external force module, and the extendible module. Besides the Hdf5 output, the I/O module adds the support of binary format in order to ensure output stability on the Sunway TaihuLight file system. The I/O module also supports data recovery from the point of interruption and computation continuity automatically. This function is expedient for large-scale simulations on supercomputers because the failure number of processors increases with the computation scale. The parallelization module involves the management functions for slave cores, which enables the master-core commands of APT-SW to assign computation tasks to slave cores. In detail, the slave-core part is integrated to be one function, and the master-core code determines when and on which slave cores the slave-core function is executed. All of the variables used in slave-core part are independent with master-core part to avoid unnecessary access of shared memory by slave cores. The communications between the LDM and shared memory happens only at the stages of initialization and finalization of slave processes. Meanwhile, the functions used by slave-core part are also optimized according to the math libraries environment of the Sunway TaihuLight supercomputer.

Figure 4 shows the schematic of master-slave acceleration of APT-SW on the Sunway TaihuLight. The  $N$  particle samplings are assigned to  $M$  master cores, each of which manages 64 slave cores. One slave core executes the iteration of one particle sample. Consequently, the total simulation time is determined by the time needed by a slave core to calculate one sampling no matter how many samplings need to be calculated. Because only the master cores can access the file system, to record the running data, we divide the  $n$  iteration steps into  $m$  groups, where  $m$  is the step number of outputs. When a master process assigns tasks to its slave processes, the data needed by iteration are copied to the LDM. Then, the 64 slave cores push their own particles for  $n/m$  steps. After that, the data of particles will be copied back to shared memory for output. As a result, the amount of communications between the master and slave core is proportional to  $m$ , and the time resolution of output data is  $T/m$ , where  $T$  is the physical time of simulation. One can change the value of  $m$  to find a balance between the time efficiency and the resolution of simulation data. For secular simulations of multi-timescale processes, the total number of iteration  $n$  can be in the magnitude order of  $10^{10}$ . It is not necessary to save all the iteration data to analyze the macroscopic behaviors. So we can set the value of  $m$  far smaller than  $n$ , which makes the time of master-slave communications ignorable compared with the iteration time. Through the procedure flow shown

in Fig. 4, APT-SW can use all the calculation capabilities of the Sunway TaihuLight and provide controllable time efficiency management. By use of APT-SW on the Sunway TaihuLight supercomputer, we have successfully completed the runaway dynamical simulations of  $10^{18}$  particle-steps which corresponds to  $10^{22}$  floating-point operations. The APT-SW thus provides an efficient platform for long-term large-scale massive-sampling simulations that can reveal the realistic physical processes more precisely.

## V. CONCLUSION

APT provides an efficient platform for large-scale particle simulations based on geometric algorithms. By tracing sampling particles, APT can precisely reveal the microscopic and macroscopic dynamical behaviors of nonlinear and complex systems, such as magnetized plasmas, even in rather complex geometries. Without extra assumptions, the simulation model of APT is the first principle model, which obeys the characteristic line equations of the Vlasov-Maxwell system. In the next version, the algorithm evaluation module will be supplied in APT to evaluate the performance of numerical methods under different complex conditions. Based on this new module, the algorithm recommendation function can help users to find the optimum algorithm for a specific problem. The graphical user interface will also be developed to make APT more convenient to use. APT will be equipped with more advanced geometric algorithms and applied to solve more pivotal scientific problems.

## ACKNOWLEDGMENTS

This research is supported by National Magnetic Confinement Fusion Energy Research Project (2015GB111003, 2014GB124005), National Natural Science Foundation of China (NSFC-11575185, 11575186, 11305171), JSPS-NRF-NSFC A3 Foresight Program (NSFC-11261140328), the CAS Key Program of Frontier Sciences (QYZDB-SSW-SYS004), and the GeoAlgorithmic Plasma Simulator (GAPS) Project.

---

[1] H. Qin and X. Guan, Phys. Rev. Lett. **100**, 035006 (2008).



- [2] H. Qin, J. Liu, J. Xiao, R. Zhang, Y. He, Y. Wang, Y. Sun, J. W. Burby, L. Ellison, and Y. Zhou, Nucl. Fusion **56**, 014001 (2015).
- [3] H. Qin, S. Zhang, J. Xiao, J. Liu, Y. Sun, and W. M. Tang, Phys. Plasmas **20**, 084503 (2013).
- [4] X. Guan, H. Qin, and N. J. Fisch, Phys. Plasmas **17**, 092502 (2010).
- [5] R. Zhang, J. Liu, H. Qin, Y. Wang, Y. He, and Y. Sun, Phys. Plasmas **22**, 044501 (2015).
- [6] J. Xiao, J. Liu, H. Qin, and Z. Yu, Phys. Plasmas **20**, 102517 (2013).
- [7] Y. He, H. Qin, Y. Sun, J. Xiao, R. Zhang, and J. Liu, Phys. Plasmas **22**, 124503 (2015).
- [8] R. Zhang, J. Liu, Y. Tang, H. Qin, J. Xiao, and B. Zhu, Phys. Plasmas **21**, 032504 (2014).
- [9] Y. He, Y. Sun, J. Liu, and H. Qin, J. Comput. Phys. **305**, 172 (2016).
- [10] Y. He, Y. Sun, J. Liu, and H. Qin, J. Comput. Phys. **281**, 135 (2015).
- [11] J. Liu, Y. Wang, and H. Qin, Nucl. Fusion **56**, 064002 (2016).
- [12] J. Liu, H. Qin, N. J. Fisch, Q. Teng, and X. Wang, Phys. Plasmas **21**, 064503 (2014).
- [13] Y. Wang, H. Qin, and J. Liu, Physics of Plasmas **23** (2016).
- [14] J. Xiao, J. Liu, H. Qin, Z. Yu, and N. Xiang, Phys. Plasmas **22**, 092305 (2015).
- [15] “Top500 website,” <https://www.top500.org/lists/2016/06/>.
- [16] H. Fu, J. Liao, J. Yang, L. Wang, Z. Song, X. Huang, C. Yang, W. Xue, F. Liu, and F. e. a. Qiao, Science China Information Sciences **59**, 072001 (2016).
- [17] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, Vol. 31 (Springer Science & Business Media, 2006).
- [18] Y. He, Y. Sun, Z. Zhou, J. Liu, and H. Qin, arXiv preprint arXiv:1509.07794 (2015).
- [19] “Lua website,” <https://www.lua.org/>.
- [20] “Hdf5 website,” <https://www.hdfgroup.org/HDF5/>.
- [21] H. Dreicer, Phys. Rev. **115**, 238 (1959).
- [22] J. Connor and R. Hastie, Nucl. Fusion **15**, 415 (1975).
- [23] R. Yoshino, T. Kondoh, Y. Neyatani, K. Itami, Y. Kawano, and N. Isei, Plasma Phys. Control. Fusion **39**, 313 (1997).
- [24] R. Jaspers, N. L. Cardozo, F. Schuller, K. Finken, T. Grewe, and G. Mank, Nucl. Fusion **36**, 367 (1996).
- [25] P. Helander, L.-G. Eriksson, and F. Andersson, Phys. Plasmas **7**, 4106 (2000).
- [26] P. Helander, L. Eriksson, and F. Andersson, Plasma Phys. Contr. Fusion **44**, B247 (2002).

- [27] T. Fülöp, H. Smith, and G. Pokol, *Phys. Plasmas* **16**, 022502 (2009).
- [28] R. Gill, B. Alper, A. Edwards, L. Ingesson, M. Johnson, and D. Ward, *Nucl. Fusion* **40**, 163 (2000).
- [29] R. Jaspers, K. Finken, G. Mank, F. Hoenen, J. Boedo, N. L. Cardozo, and F. Schuller, *Nucl. Fusion* **33**, 1775 (1993).
- [30] R. Nygren, T. Lutz, D. Walsh, G. Martin, M. Chatelier, T. Loarer, and D. Guilhem, *J. Nucl. Mater.* **241**, 522 (1997).
- [31] P. Parks, M. Rosenbluth, and S. Putvinski, *Phys. Plasmas* **6**, 2523 (1999).
- [32] M. Rosenbluth and S. Putvinski, *Nucl. Fusion* **37**, 1355 (1997).
- [33] R. Yoshino and S. Tokuda, *Nucl. Fusion* **40**, 1293 (2000).
- [34] H. Tamai, R. Yoshino, S. Tokuda, G. Kurita, Y. Neyatani, M. Bakhtiari, R. Khayrutdinov, V. Lukash, and M. Rosenbluth, *Nucl. Fusion* **42**, 290 (2002).
- [35] M. Lehnen, S. Bozhnikov, S. Abdullaev, and M. Jakubowski, *Phys. Rev. Lett.* **100**, 255003 (2008).
- [36] K. Finken, S. Abdullaev, M. Jakubowski, R. Jaspers, M. Lehnen, R. Schlickeiser, K. Spatschek, A. Wingen, and R. Wolf, *Nucl. Fusion* **47**, 91 (2007).
- [37] N. J. Fisch, *Rev. Mod. Phys.* **59**, 175 (1987).
- [38] H.-W. Bartels, *Fusion Eng. Des.* **23**, 323 (1994).
- [39] T. Kawamura, H. Obayashi, and A. Miyahara, *Fusion Eng. Des.* **9**, 39 (1989).
- [40] H. Bolt, A. Miyahara, M. Miyake, and T. Yamamoto, *J. Nucl. Mater.* **151**, 48 (1987).
- [41] R. Jaspers, N. L. Cardozo, A. Donne, H. Widdershoven, and K. Finken, *Rev. Sci. Instrum.* **72**, 466 (2001).
- [42] J. Martín-Solís, J. Alvarez, R. Sánchez, and B. Esposito, *Phys. Plasmas* **5**, 2370 (1998).
- [43] J. Martín-Solís, B. Esposito, R. Sánchez, and J. Alvarez, *Phys. Plasmas* **6**, 238 (1999).
- [44] J. A. Van Allen and L. A. Frank, *Nature* **183** (1959).
- [45] Y. Chen, G. D. Reeves, and R. H. Friedel, *Nat. Phys.* **3**, 614 (2007).
- [46] G. Reeves, H. E. Spence, M. Henderson, S. Morley, R. Friedel, H. Funsten, D. Baker, S. Kanekal, J. Blake, and J. Fennell, *Science* **341**, 991 (2013).